ISSN: 0828-3494
ISBN: 0-7731-0447-X

# Discovering Causality and Acausality in Temporal Data

Kamran Karimi & Howard J. Hamilton

Technical Report CS-2003-02
February, 2003


Department of Computer Science
University of Regina
Regina, Saskatchewan
CANADA  S4S 0A2

# Discovering Causality and Acausality in Temporal Data

Kamran Karimi & Howard J. Hamilton
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
{karimi, hamilton}@cs.uregina.ca

## Abstract

The problem of determining whether or not the value of an attribute is caused by other observed attributes, or they merely happen to occur together, has been attacked from different angles. In this paper we propose a solution to the problem of distinguishing between causal and acausal temporal rules, and the system that generated the rules. The proposed method, called the Temporal Investigation Method for Enregistered Record Sequences (TIMERS) is explained and introduced formally. TIMERS assumes that time can flow in two directions, forward, which is the natural flow, and backward. This method has been implemented in the TimeSleuth software. We assume that the input to TIMERS consists of a sequence of records, where each record is observed at regular intervals. A set of rules is then generated from this input data. We perform three tests. One to determine if the set of rules describes an instantaneous relationship, where the decision attribute depends on condition attributes seen at the same time instant. The other two tests determine the degree to which a set of rules is causal or acausal by changing the direction time when generating temporal rules. The results of the three tests are then used to declare a verdict as to the nature of the system: Instantaneous, causal, or acausal. Unlike approaches based on causal Bayesian networks, our approach does not emphasize relations among individual attributes. Its verdict applies to a set of rules, and the system that has generated the temporal input data.

## 1. Introduction

When a certain value of a decision attribute is often seen together with certain values of some condition attributes, the question may arise as to whether there is a causal relationship between the decision and condition attributes. This problem becomes harder when we consider a system in which the current decision attribute may have been determined by previous values of the condition attributes and the previous values of the decision attribute itself. Distinguishing among causal and acausal systems is important because a causal system can be controlled if the input can be manipulated, and if we know the relationships between the input values and the output value. An acausal system may not allow the output to be determined in this way. Even though the previous values may always precede the decision attribute, The real cause is probably hidden. Thus reproducing a desired output is not guaranteed. This insight is very useful to a domain expert who wants to discover the characteristics of a system under investigation.

A popular method for assessing the causality of a relationship is to use statistical methods and determine how two attributes influence each other [8]. A test for conditional independence is performed, which says that if $P(y, z \mid x) = P(y \mid x)$, then $y$ is conditionally independent from z given $x$. If we are building a causal tree, this independence would be represented by $x$ being a parent of $y$, and $z$ being either a parent of $x$, or residing in another branch of the tree. This method thus works by first assuming that all attributes are dependent on all other attributes. Tests for conditional independence prune some of these dependencies, and those that remain are considered to show the causal relationships among the attributes.

We consider time to be of extreme importance in the discussion of causality. In physics this distinction may not be made, as many relations seem to be instantaneous. For example, consider the formula $PV = nRT$, where $P$ is pressure, $V$ is volume, $n$ and R are considered constants, and $T$ is temperature. Setting the volume as constant, increasing temperature will accompany an increase in pressure with no time delay. A higher temperature increases the speed at which the molecules move, which is the same reason for the increasing pressure value. In other words, temperature and pressure measure the same property of the system. For us, the problem to solve is when there is an appreciable delay between an action and the results. For example, setting the temperature gauge to a higher value will take time to have an effect on the temperature (and pressure) of the system. Suppose our system consists of a closed container filled with some gas, heated by a heater. The problem we are interested in is determining whether there is a relation between changing the temperature of the heater on one hand, and the gas pressure on the other hand. At the start of the experiment, at time 0, we determine the pressure of the system to have be $P_0$, the temperature of the system to be $T_0$, and the heater's temperature setting to be 500 C. We change the heater's temperature to 600C. After a while, at time 1, we determine the pressure to be $P_1$, the temperature to be $T_1$, and the heater's setting to be 600C as expected. The question we want answered is whether there is a relation among the heater setting and the temperature and pressure of the system. In the rest of the paper $T$ will always denote time.

In previous work we have looked at other methods of discovering causality, such as TETRAD [10] and CaMML [6]. One important property that differentiates the method we will present in this paper from these is that the presented method deals with data originating from the same source over time, while others deal with data generated from different sources with no special temporal ordering. We have shown that when dealing with the appropriate kind of data, the presented method performs well [5]. Our previous work was concerned with discovering temporal rules, with no consideration of causality and acausality. Here we introduce an extension to our method to aid a domain expert in making that distinction.

In this paper we introduce the TIMERS method (Temporal Investigation Method for Enregistered Record Sequences). As already mentioned, it is different from the other methods in a number of ways. First, this method is based on an explicit temporal order among the observed values of the attributes. Suitable input consists of an ordered set of records, where each record contains the values of the attributes, all observed at the same time. An example of such a record is: $<x = 1, y = 2, z = 3>$. These records should have been registered at regular intervals. The other difference is that this method is not concerned about the relation among individual attributes, such as "*x* and y are causes of *z*." Instead, it judges a *set of temporal rules* that involves the values of *x* and *y* to predict the value of *z*, as being either causal or acausal. An example rule that could belong to this rule set is: **if** $\{(x = 1)$ **and** $(y = 2)\}$ **then** $(z = 3)$. Here *x* and y are considered to be condition attributes, while *z* is the decision attribute. By itself, we cannot tell if this rule represents a causal relation or not, i.e., do *x* and *y* cause *z* to have a certain value, or they just happen to be seen together, and all their values are caused by some hidden variable(s). The TimeSleuth software [2], written in Java and freely available, implements the TIMERS method and tries to answer this question. This method is especially appropriate when we have access to many attributes of a system, because the more attributes we have, the better the chances of finding meaningful relationships among them. Otherwise there is a danger of finding random relations that happen to exist in the sample of data used for the causality investigation. For example, in weather data having only the air pressure, wind direction, and wind speed, trying to find a set of causal/acausal rules to predict the value of any of these attributes using the other two may not lead to satisfactory results because of the lack of relevant information. However, if we also had information such as the air temperature, soil temperature, humidity, cloud coverage, etc. then the method might perform better.

The rest of the paper is outlined as follows. Section 2 defines the two directions for time, forward and backward, and describes an operation called flattening, to bring the relevant attributes together with different directions of time. Section 3 formally defines causality and acausality in the context of the TIMERS method. The distinction among temporal and atemporal rules is also made clear. Section 4 explains how TIMERS determines the nature of a set of rules. We explain our assumption that in a temporal environment, a forward flow of time can be used to discover causality, and a backward direction can be used as a test for acausality. Section 5 presents the results of experiments performed with the TimeSleuth software using real and synthetic data sets. Section 6 concludes the paper.

## 2. Forward and Backward Directions of Time

A temporal rule is one that involves variables from times different than the decision attribute's time of observation. An example temporal rule is:

**If** {(At time $T_{-3}$: $x = 2$) **and** (At time $T_{-1}$: $y > 1, x = 2$)} **then** (At time $T$: $x = 5$). (Rule 1).

This rule indicates that the current value of $x$ (at time $T$ ) depends on the value of $x$, 3 time steps ago, and also on the value of $x$ and $y$, 1 time step ago. We use a preprocessing technique called flattening [3] to change the input data into a suitable form for extracting temporal rules with tools that are not based on an explicit representation of time. With flattening, data from consecutive time steps are put into the same record, so if in two consecutive time steps we have observed the values of $x$ and $y$ as: Time $n$: $<x = 1, y = 2>$, Time $n + 1$: $<x = 3, y = 2>$, then we can flatten these two records to obtain <Time $T$ - 1: $x_1 = 1$, $y_1 = 2$, Time $T$: $x_2 = 3$, $y_2 = 2$>. The "Time <number>" keywords are implied, and do not appear in the records. The initial temporal order of the records is lost in the flattened records, and time always starts from ($-w$ - 1) or ($T$ - $w$ - 1) inside each flattened record, and goes on until 0 or $T$. 0 or $T$ signifies the "current time" which is relative to the start of each record. Such a record can be used to predict the value of either $x_2$ or $y_2$ using the other attributes. Since e refrain from using any condition attribute from the current time, we modify the previous record by omitting either $x_2$ or $y_2$.

In the previous example we used *forward flattening*, because the data is flattened in the same direction as the forward flow of time. We used the previous observations to predict the value of the decision attribute. The other way to flatten the data is *backward flattening*, which goes against the natural flow of time. Given the two previous example records, the result of a backward flattening would be < Time $T$: $y_1 = 2$, Time $T + 1$: $x_2 = 3$, $y_2 = 2$>. Inside the record time starts at 0 or $T$, and ends at ($w$ - 1) or ($T + w$ - 1). This record could be used to predict the value of $y_1$ based on the other attributes. $x_1$ is omitted because it appears at the same time as the decision attribute $y_1$. In the backward direction, *future* observations are used to predict the value of the decision attribute.

Flattening, in either direction brings data from different time steps together. Rules found using such flattened records are of temporal nature. The output set of rules distinguishes among the attributes from different times by using time tags, as shown in the example rule above. If the system under investigation contains causal relations, then bringing possible causes and effects from different time steps together allows TIMERS to find causal rules in the flattened records. The number of records flattened together is determined by the window size. In the above example, the window size is 2. Different window sizes can be tried to see which value gives the best results.

Given a set of $N$ temporally ordered observed records $\boldsymbol{D} = \{\boldsymbol{rec}_1, \ldots, \boldsymbol{rec}_N\}$, the problem is to find a set of rules, as described in more detail below. Each record $\boldsymbol{rec}_t = <c_{t1}, \ldots, c_{tm}>$ gives the values of a set of variables $V = \{v_1, \ldots, v_m\}$ observed at time step $t$. . The *forward window set* $P_f(w, t) = \{d_t, c_{ki} \mid (w \le t) \;\&\; t\text{-}w+1 \le k < t, 1 \le i \le m\}$ represents all observations in the window of size $w$, starting at time ($t$ - $w$ +1) and going until time $t$, which is considered the current time. Time flows forward, in the sense that the decision attribute appears at the end (time $t$). $d_t$ is the decision attribute at time $t$. The *backward window set* $P_b(w, t) = \{ d_t, c_{ki} \mid (t \le |\boldsymbol{D}| - w + 1) \;\&\; t < k \le (t + w - 1), 1 \le i \le m\}$ represents all observations in the window starting at time $t$ and going until time ($t + w$ - 1). Again, $t$ is considered to be the current time. Time flows backward, in the sense that the decision attribute appears at the beginning (time $t$). At the time step containing the decision attribute, condition attributes do not appear. In other words, $d_t$ is the only variable at current time $t$.

Formally, the flattening operator $F(\boldsymbol{w}, \boldsymbol{D}, \boldsymbol{direction}, \boldsymbol{d})$ takes as input a window size $w$, the input records $\boldsymbol{D}$, a time direction ***direction***, and the decision attribute $\boldsymbol{d}$, and outputs flattened records according to the algorithm in Figure 1.

$F(\boldsymbol{w}, \boldsymbol{D}, \boldsymbol{direction}, \boldsymbol{d})$
Begin
    for ($t = 1$ to $|\boldsymbol{D}|$)
    begin

        if (($\boldsymbol{direction} = forward$) and ($t \geq \boldsymbol{w}$))
            output ($\boldsymbol{z} = \langle z_{ki} \mid d_{pi} \in P_f(\boldsymbol{w}, t)$ & $k = w\text{-}1\text{-}t{+}p$ & $z_{ki} = d_{pi}\rangle$)

        else if (($\boldsymbol{direction} = backward$) and ($|\boldsymbol{D}| - \boldsymbol{w} + 1 \geq t$))
            output ($\boldsymbol{z} = \langle z_{ki} \mid d_{pi} \in P_b(\boldsymbol{w}, t)$ & $k = p - t$ & $z_{ki} = d_{pi}\rangle$)

    end
end

Figure 1.The flattening operation. The decision attribute d is used by the $P_f()$ and $P_b()$ sets.

The flattened record contains the neighboring $w$ records in the appropriate direction of time. The $F_{\mathrm{w}}$ operator renames the time index values so that in each record, time is measured relative to the start of that record only. In each flattened record, the time index ranges from 0 to $w$-1. The flattened records are thus independent of the time variable $t$, making time relative to the start of each flattened record.

Each rule $r$, generated from these flattened records is a pair. The first member of a rule is a set of tests. The other member of the rule is the value that is predicted for the decision variable at time 0 or $w$-1. $r = (Tests_r, d_{\mathrm{val}})$, where $Tests_r = \{ Test = (a, x, Cond) \}$. Where $a \in V$, and $x$ is the time in which the variable $a$ appears, and where $Cond$ represents the condition under which $Test$ succeeds. One example is: $a_{\mathrm{x}} > 5$. $d_{\mathrm{val}}$ is the value predicted for $d_{\mathrm{t}}$ (the decision attribute at time t).

For subsequent discussions, we define the operator CONDITION(r) as the set of variables that appear in the condition side of a rule, i.e., CONDITION = $\{a_{\mathrm{x}} \mid (a, x, Cond) \in Tests_{\mathrm{r}}\}$. Similarly, we define DECISION(r) = $\{d_{\{0,\, w\text{-}1\}}\}$.

An *aggregate* variable is a synthetic variable that is computed by applying a function to the value of a condition variable over a window size. An example is the average value of a condition variable. An aggregate variable is assumed to have happened at all of the time steps in the window. An aggregate variable is supposed to have happened before (in case of forward time flow) or after (in case of backward time flow) the decision attribute, and its presence does not invalidate a temporal rule.

## 3. Temporal Causality and Temporal Acausality

There is no consensus on the definitions of terms like causality or acausality. For this reason we provide our own definitions here. In previous research we detected sets of temporal rules and assigned the task of whether such a relationship is causal to a domain expert [4]. Here we provide a way to make such distinction. Even though TIMERS provides an algorithmic method for making a decision through a set of metrics, a domain expert is still making the final decision. In each case, the informal definition is followed by a formal one. We consider a set of rules to define a relationship among the condition attributes and the decision attribute.

### 3.1 Instantaneous

An *instantaneous* set of rules is one in which the current value of the decision attribute relies solely on the current values of the condition attributes in each rule [11]. An instantaneous set of rules is an *atemporal* one. Another name for an instantaneous set of rules is a (atemporal) *co-occurrence*, where the values of the decision attribute is associated with the values of the condition attributes.

**Definition 1.** For any given rule $r$ in the rule set $R$, if the decision attribute $d$ appears at time $T$, then all condition attributes should also appear at time $T$:
$R$ is instantaneous iff ($\forall\ r \in R$, if $d_T$ = DECISION($r$), then $\forall\ a_t \in$ CONDITION($r$), $t = T$ ).

### 3.2 Temporal

A *temporal* set of rules is one that involves attributes from different time steps. A temporal set of rules can be causal or acausal.

**Definition 2.** For any rule $r$ in the rule set $R$, if the decision attribute appears at time $T$, then all condition attributes should appear at time $t \neq T$.
$R$ is temporal iff ($\forall\ r \in R$, if $d_T$ = DECISION($r$), then $\forall\ a_t \in$ CONDITION($r$), $t \neq T$ ).

We do not include the current time in the definition of a temporal set of rules because doing so can make it possible for a co-occurrence relation to "pollute" the results of causality and acausality tests, which are introduced later. As an example, consider the case of a co-occurrence relationship among three attributes: the $x$ and $y$ positions as the condition attributes, and the presence of food at that location as the decision attribute. If food always exists in the same location, then we can predict the presence of food even when we are adjacent to the food location. If we move from position to position as time passes, then temporal information is as good as atemporal information, because they provide the same information. To prevent this from happening, we refrain from using observations that happen at the same time as the decision attribute.

We now define the two possible types of a temporal rule:

#### 3.2.1 Causal

In a *causal* set of rules, the current value of the decision attribute relies only on the previous values of the condition attributes in each rule [11].

**Definition 3.** For any rule $r$ in the rule set $R$, if the decision attribute $d$ appears at time $T$, then all condition attributes should appear at time $t < T$.
$R$ is causal iff ($\forall\ r \in R$, if $d_T$ = DECISION($r$), then $\forall\ a_t \in$ CONDITION($r$), $t < T$ ).

#### 3.2.2 Acausal

In an *acausal* set of rules, the current value of the decision attribute relies only on the future values of the condition attributes in each rule [7].

**Definition 4.** For any rule $r$ in the rule set $R$, if the decision attribute $d$ appears at time $T$, then all condition attributes should appear at time $t > T$.
$R$ is acausal iff ($\forall\ r \in R$, if $d_T$ = DECISION($r$), then $\forall\ a_t \in$ CONDITION($r$), $t > T$ ).

All rules in a causal rule set have the same direction of time, and there are no attributes from the same time as the decision attribute. This property is guaranteed simply by not using condition attributes from the same time step as the decision attribute, and also by sorting the condition attributes in an increasing temporal order, until we get to the decision attribute. The same property holds for acausal rule sets, where time flows backward in all rules till we get to the decision attribute. Complementary, in an instantaneous rule set, no condition attribute from other times can ever appear.

## 4. The TIMERS Method

The TIMERS method is based on finding classification rules to predict the value of a decision attribute using a number of condition attributes that may have been observed at different times. We extract different sets of rules to predict the value of a condition attribute based on different window sizes and different directions for the flow of time. The quality of the set of rules determines how the window size and time direction have been appropriate. We choose either the training accuracy or the predictive accuracy of the set of rules as the metric for a good window size. The training accuracy measures the applicability of all the discovered rules on the data that generated the rules, and is intuitive and easy to compute. The predictive accuracy tries the generated rules on unseen cases, and measures the predictive power of the rules. While predictive accuracy can only be generated if there are unseen cases, the training accuracy is always available.

Flattening data results in a test for causality because we ask C4.5 to find rules to predict the value of the decision attribute using previously observed condition attributes. TimeSleuth can as easily test for the acausality by flattening the data in the reverse temporal order. Thus the example record in the previous section would be flattened to generate $<x_2 = 3, y_2 = 2, x_1 = 2>$ which could be used to predict the value of either $x_1$ using the other attributes. We would thus test to see if the current value of a decision attribute depends on the values of other attributes in the future. In practice, is not necessary to write out the records in the reverse order, as long as it is clear which variable is the decision attribute. In the forward flow of time, it is either $x_2$ or $y_2$ while in the reverse flow of time it is either $x_1$ or $y_1$.

To test to see if a set of rules is spontaneous, we simply refrain from flattening the input. In other words, we use a window size of $w = 1$. We thus test to see if the current value of the decision attribute depends on the current value of the condition attribute.

As mentioned before, in TIMERS we use either the training or the predictive accuracy as the measure for the quality of the set of rules. C4.5 being a classifier, does its best to come up with accurate rules with no regard to the way the input has been preprocessed. The user should thus perform 3 tests, generates different rule sets, and compares the results with each other in order to judge the system as causal, acausal, or instantaneous.

We provide the following guideline for making the distinction among the three types. The tests are done in the order presented:

Generate data with the data and with different values of the window size. Set the accuracy of the rule set for $w = 1$ to $a_i$ (instantaneous), set $a_f$ to the best accuracy value derived from forward flow (causal) tests, and set $a_b$ to the best value for backward flow (acausal) tests. Consider RuleGenerator() to be a function that takes as input the flattened records and a target decision attribute, generates temporal rules, and returns a quality measure for the rules, such as their training or testing accuracy.

$a_i$ = RuleGenerator($D$, $d$)
$a_f$ = max($1 < \alpha \leq w \leq \beta$, RuleGenerator(Flat($w$, $D$, *forward*, $d$), $d$))
$a_b$ = max($1 < \alpha \leq w \leq \beta$, RuleGenerator(Flat($w$, $D$, *backward*, $d$), $d$))

After obtaining the above values, we change the focus from sets of rules to the system that generated the input data, and follow these guidelines to decide on the nature of the system under investigation.

1. If the results of an instantaneous test (the test with a window size of 1) is better than other tests, then the system is spontaneous. If a system is spontaneous, we expect the results of temporal (causal and acausal) tests to be about the same as the spontaneous test, as information available in flattened records will contain co-occurrences among the attributes.
2. If the results of a backward flow test is better than a spontaneity test, and better or equal to a causality test, then we declare the rules to denote an acausal relation. We opt to declare the rule set acausal in case of a tie because judging a system as causal needs insight into the way the system works, and it may not be available to our method.
3. If the results of a forward flow test are better than the other two tests, the rules are causal. Making the decision to declare a system as causal as the last choice reflects our conservative approach.

If $((a_i \geq_\varepsilon a_f) \wedge (a_i \geq_\varepsilon a_b))$ then the system is instantaneous.
Else if $(a_b \geq_\varepsilon a_f)$ then the system in acausal
Else the system is causal.

The $\varepsilon$ subscripts in the comparison operators capture the "about" phrase in the informal representation of the method. $a >_\varepsilon b$ is defined as: $a > b + \varepsilon$. Similarly, $a \geq_\varepsilon b$ is defined as $a \geq b + \varepsilon$. The value of $\varepsilon$ is determined by the domain expert.

We assume that once the domain expert finds a good training or predictive accuracy in either temporal direction and with any window size, he will employ the corresponding rule set to make predictions about the decision attribute. For example, if among 10 window values, the best result is obtained in one backward flow test, while in the other 9 tests the results of a forward flow of time are about the same or only slightly better than the results of a backward flow of time at the corresponding window size, we still mark the system as acausal in that window size range. The assumption being that the one best rule set obtained with a backward flow will be utilized.

Alternatively, we can talk about the nature of a system at a specific window size $w$. In this case, we follow the previous steps, but we use
$a_f$ = RuleGenerator(Flat($w$, $D$, *forward, d*), $d$)
$a_b$ = RuleGenerator(Flat($w$, $D$, *backward, d*), $d$)

In such a case, we say the system is causal or acausal, at window size $w$.

To safeguard against cases where the amount of information is not sufficient, we consider it necessary for the domain expert to have thresholds for how low the accuracy in the tests could be before the results are discarded. For example, if the three tests all result in accuracy values less than 20%, then we could conclude that the data is insufficient for making any judgment. The TIMERS Algorithm is presented in Figure 2.

**TIMERS**($D$, $\alpha$ , $\beta$, $A_{th}$ , $\varepsilon$, d)
**Input:** A sequence of temporally ordered data records $D$, a minimum and maximum flattening window size $\alpha$ and $\beta$, $\alpha \leq \beta$. A minimum accuracy threshold $A_{th}$, a tolerance value $\varepsilon$, and a decision attribute $d$. The attribute $d$ can be set to any of the observable attributes in the system, or the algorithm can be tried on all available attributes in turn.

**Output:** A verdict as to whether the system behaves in a instantaneous, causal or acausal manner when predicting the value of a specified decision attribute.

RuleGenerator() is a function the receives input records, generates decision rules, and returns the training or predictive accuracy of the rules.

Flat($w$, $D$, *TimeFlow,* $d$) is a function that flattens the records in dataset $D$ in the forward (causal) or backward (acausal) direction, depending on the value of *TimeFlow*, the window size $w$, the decision attribute $d$, and returns the flattened records.

**Begin:**
$a_i$ = RuleGenerator($D$, $d$);
for ($w = \alpha$ to $\beta$)
  $a_{fw}$ = RuleGenerator(Flat($w$, $D$, *forward,* $d$), $d$)
  $a_{bw}$ = RuleGenerator(Flat($w$, $D$, *backward,* $d$), $d$)
 end for

$a_f = \max(a_{f\alpha},.. a_{f\beta})$
$a_b = \max(a_{b\alpha},\ldots a_{b\beta})$

if $((a_i < A_{th}) \wedge (a_f < A_{th}) \wedge (a_b < A_{th}))$ then discard results and stop. Not enough information to make a verdict.

if $((a_i >_\varepsilon a_f) \wedge (a_i >_\varepsilon a_b))$ then verdict = "the system is instantaneous"
else if $(a_f \geq_\varepsilon a_b)$ then verdict = "the system is acausal"
else verdict = "the system is causal"

if $w_1 = w_2$ and verdict ≠ "the system is instantaneous" then verdict = verdict + "at window size" + $w_1$

verdict = "for attribute $d$, " + verdict

Return the verdict.
**End.**

Figure 2. The TIMERS algorithm, performed for the decision attribute $d$.

## 5. Experimental results.

In this section we try the proposed method, as implemented in TimeSleuth. It is an application software written in Java, and usable anywhere a Java runtime system and a graphical user interface is available. C4.5T, a modified form of the standard C4.5 [9] that is able to derive temporal rules, is used as the rule generator. C4.5T's output includes temporal information. An example temporal rule output by C4.5T is Rule 1 is Section 3.

We use three different data sets, each with its own characteristics. The first is synthetic artificial life data, generated from a simulated world. The second data set comes from a weather database, with relevant fields, making it possible for the rule generator to find reliable rules. The third one is weather data with few relevant condition attributes, where it is hard to find useful rules.

**Series 1**: The first series of experiments used data set from an artificial life program called URAL [13]. In a two-dimensional world, a robot moves around randomly, and records its current location plus the action that will take it to the next ($x$, $y$) position. It also records the presence of food at each location. The program chooses fix locations for the food, and keeps them there all through the run of the program. The robot can move to left and right along the $x$ axis, and also up and down, along the $y$ axis, with no problems, except for the edges of the world, where a move will leave it in the same position. The next position is thus reliably predictable if we know the current position and the direction that the robot will take at the position. This is a temporal relationship, as the robot makes one move in every clock tick, making the current position dependent of the previous position and movement direction. Food exists on fixed location, and at each location, the presence of food does not depend on any previous moves or positions taken by the robot.

We first set the position along the $x$ axis as the decision attribute. In our world, the current $x$ value depend of the previous $x$ value and the previous movement direction. The results appear in Table 1.

| Window Size | Causality Test | Acausality Test |
|---|---|---|
| 1 | **46.0%** | |
| 2 | **100%** | 70.6% |
| 3 | 100% | 71.7% |
| 4 | 100% | 72.8% |
| 5 | 100% | **74.5%** |
| 6 | 100% | 73.7% |
| 7 | 100% | 73.3% |
| 8 | 100% | 73.6% |
| 9 | 100% | 72.6% |
| 10 | 100% | 73.9% |

Table 1. URAL data. Decision attribute is $x$.

The system is not instantaneous, because a window size of 1 (current time) gives relatively poor results. Rather, the system is causal, because the forward test gives better relative results. The same conclusions are obtained for the $y$ values.

In the following table we set food as the decision attribute. In this simple world, the presence of food is associated with its position. Because the location of the food does not change, we can predict the presence of food using only the positions that neighbour the food. We expect about the same results with any value of window size.

| Window Size | Causality Test | Acausality Test |
|---|---|---|
| 1 | **99.5%** | |
| 2 | 99.3% | 99.1% |
| 3 | 99.3% | 99.3% |
| 4 | 99.3% | 99.4% |
| 5 | 99.4% | 99.5% |
| 6 | 99.4% | **99.6%** |
| 7 | 99.4% | 99.5% |
| 8 | **99.5%** | 99.5% |
| 9 | 99.3% | 99.6% |
| 10 | 99.3% | 99.6% |

Table 2. URAL data. Decision attribute is presence of food

The system is instantaneous because a window size of 1 gives relatively good results.

**Series 2:** The second experiment was done on a real-world data set, comprising Louisiana weather observations [12]. The observed attributes consist of air temperature, amount of rain, maximum wind speed, average wind speed, wind direction, humidity, solar radiation, and soil temperature. These values were recorded every hour. We used 343 consecutive observations to predict the value of the attribute soil temperature.

| Window Size | Causality Test | Acausality Test |
|---|---|---|
| 1 | 27.7% | |
| 2 | 82.7% | 75.1% |
| 3 | **86.8%** | **87.1%** |
| 4 | 84.4% | 84.7% |
| 5 | 86.7% | 82.9% |
| 6 | 77.5% | 81.4% |
| 7 | 79.5% | 79.8% |
| 8 | 80.7% | 79.8% |
| 9 | 77.9% | 77.3% |
| 10 | 79.2% | 74.0% |

Table 3. Louisiana data. Decision attribute is Soil temperature.

The system is not instantaneous, because a window size of 1 gives poor results. The system is acausal, since the acausality tests gives the same or better results that the causality test.

**Series 3:** The next test was done on the Helgoland weather data set [1], which consists of hourly observations of the following attributes: year, month, day, hour, air pressure, wind direction, and wind speed. The decision attribute was set to be the wind speed. 3000 hours of consecutive observations were used to produce the results.

| Window Size | Causality Test | Acausality Test |
|---|---|---|
| 1 | **18.9%** | |
| 2 | **17.7%** | **20.7%** |
| 3 | 14.7% | 17.2% |
| 4 | 14.2% | 16.9% |
| 5 | 13.9% | 14.5% |
| 6 | 14.0% | 15.2% |
| 7 | 13.4% | 15.0% |
| 8 | 13.2% | 14.9% |
| 9 | 12.2% | 13.9% |
| 10 | 12.0% | 14.7% |

Table 4. Helgoland data. Decision attribute is Wind speed.

There is not enough information to make a judgement. In spite of the thousands of records, each record is not rich enough to provide TimeSleuth with a reliable way of creating rules to predict the value of the decision attribute. This is unlike the previous weather data set, and is clearly reflected in the accuracy of the rule, which is low.

## 6. Concluding Remarks

We introduced the TIMERS method to judge the nature of the relationship among a set of condition attributes and a decision attribute. The three possibilities are: Instantaneous, causal, and acausal. The verdict is determined by examining sets of rules that predicts the value of the decision attribute. TIMERS relies on temporal characteristics of the rules, and uses the quality of the rules generated under different directions of time as a criterion to determine if there is a causal or acausal relationship in the system that generated the input data. TIMERS was presented formally, and intuitions about our choices were also provided as needed. TimeSleuth was introduced as the software that implements the TIMERS method.

We applied TimeSleuth to three different data sets. The first data set came from an artificial world, where the nature of the relationships were clear and there are no exceptions to the rules. In this ideal environment, TimeSleuth performed very well. We then moved to the real-world problem of the Louisiana weather observations, where the data set consisted of many related attributes. TimeSleuth could judge the nature of the rules to predict the value of the soil temperature because there was enough relevant information in the data and the rules. In the third set of data, TimeSleuth was tried on a "long but narrow" weather data, where the attributes were not relevant enough to predict the value of the attribute "wind speed." This was reflected in low accuracy values. This data set is probably more suitable for a time-series analysis.

TimeSleuth is written in Java and freely available by contacting the authors or from the address http://www.cs.uregina.ca/~karimi/downloads.html. It includes sources, executables, example files, and online help.

## References

[1] Hoeppner, F., Discovery of Temporal Patterns: Learning Rules about the Qualitative Behaviour of Time Series, *Principles of Data Mining and Knowledge Discovery* (*PKDD'2001*)*,* 2001.
[2] Karimi, K., and Hamilton, H.J., TimeSleuth: A Tool for Discovering Causal and Temporal Rules, *14th IEEE International Conference On Tools with Artificial Intelligence (ICTAI2002)*, Washington DC, USA, November 2002.
[3] Karimi K., and Hamilton H.J., Learning With C4.5 in a Situation Calculus Domain,The Twentieth SGES International Conference on Knowledge Based Systems and
Applied Artificial Intelligence (ES2000), Cambridge, UK, December 2000.
[4] Karimi K., and Hamilton H.J., Discovering Temporal Rules from Temporally Ordered Data, The Third International Conference on Intelligent Data Engineering and
Automated Learning (IDEAL'2002), Manchester, UK, August 2002.
[5] Karimi, K. and Hamilton, H.J., Finding Temporal Relations: Causal Bayesian Networks vs. C4.5, *The Twelfth International Symposium on Methodologies for Intelligent Systems (ISMIS'2000)*, Charlotte, NC, USA, October 2000.
[6] Kennett, R.J., Korb, K.B., and Nicholson, A.E., Seabreeze Prediction Using Bayesian Networks: A Case Study, *Proc. Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01).* Hong Kong, April 2001.
[7] Krener, A. J. Acausal Realization Theory, Part I; Linear Deterministic Systems. *SIAM Journal on Control and Optimization*. 1987. Vol 25, No 3, pp 499-525.
[8] Pearl, J., *Causality: Models, Reasoning, and Inference*, Cambridge University Press. 2000.
[9] Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
[10] Scheines, R., Spirtes, P., Glymour, C. and Meek, C., *Tetrad II: Tools for Causal Modeling*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.
[11] Schwarz, R. J. and Friedland B., *Linear Systems*. McGraw-Hill, New York. 1965.
[12] http://typhoon.bae.lsu.edu/datatabl/current/sugcurrh.html. Contents varies.
[13] http://www.cs.uregina.ca/~karimi/downloads.html/URAL.java